W.A.C.O. PRINTOUT

#29 SEPT, 1987

--The G-Man--

You, the Law and the Truth

by David G. Grace
Special Agent for the United States

As a subscriber to three computer magazines, user of one commercial on-line service, caller of dozens of local boards, avid purchaser of software and a member of W.A.C.O., I have noticed the increased interest in software and hardware piracy. We are becoming more aware of Copyright warning labels which seem to quote the law and warn of its wrath. These labels are sometimes titled "FBI Warning" as if that agency had placed it there themselves. Even the low-key approach of some anti-piracy advertisements by Software Associations can be misleading. Whether in person or print, the appearance of legal jargon has found its way into the discussions which had previously dealt only with the economic and moral issues.

There is, of course, nothing wrong with information concerning the legalities of certain actions being available to all concerned. But I object to the source being a computer programmer, market manager or commercial association. After all, if you were stuck with a particularly nasty program bug, would your first call be to a lawyer? Most computer-related information exchanged during the breaks at a user group meeting is both factual and helpful. But the legalities of piracy, mainly what is and isn't a criminal violation, are spread with the speed and accuracy of a schoolyard sex lesson. Let's cut through the crap, for once, and see who makes the decisions, how they're made and what does the law really say. You may be surprised.

Copyright violation is a Federal crime and is, of course, prosecuted in the Federal court system. But who ultimately decides who may be convicted of any Federal crime? Not the Jury, they only determine guilt or innocence based on the facts which they are permitted to receive. Not the Judge (or even the Supreme Court), they only referee and rule on points of law and the Constitution. Not the Grand Jury, they vote whether to indict based almost exclusively on the Government's testimony. Not the Federal Agent, he only presents the result of his investigation.

To be sure, if the Agent didn't investigate, if the Grand Jury didn't indict, if the Judge dismissed the case, if the Jury found the Defendant Not Guilty, or if the Supreme Court found the law (or its enforcement) Unconstitutional, then there would be no conviction. But it is the United States Attorney (the District Attorney of the Federal court system) who must actively seek prosecution of the case, if there is to be any prosecution at all. Like the D.A., he receives the result of the law enforcement officer's investigation and makes the all-powerful decision.

Also like the D.A., the decision is often delegated to his staff of Assistant United States Attorneys who actually prosecute the cases. The A.U.S.A. often specializes in prosecuting a specific set of laws, like those dealing with Copyrights. They make their decisions after answering three basic questions: "Can I win this case? Does this case have sex appeal? Will the Defendant plead Guilty?" He won't take a case he is unlikely to win because he wishes to preserve his Won-Lost record. The sex appeal is for the Jury, the media and/or the U.S. Attorney, because nobody wants to try a case which makes the Government look like a bully in court or the Defendant look like the downtrod peasant. Of course, if the Defendant will enter a Guilty plea, the other two questions need not be asked!

Now that we've identified the primary decision-maker, let's assume that you are "worthy" of Federal prosecution. The one question that may be of general interest is, "Sure, software piracy is a Federal crime. But what's the bottom line? Where does one cross the line from technical violation to official Defendant status? There are three basic elements to the Federal crimes concerning Copyright violation. The subject material must be copyrighted, it must be copied (and/or sold, distributed, etc.), and the motive must be profit. Now this profit may be financial gain, or just about any tangible or intangible profit you can name. In order for a Federal violation of law to take place, all three elements must be present. Since Federal law does not discriminate between types of Copyrighted material (in this general section), this would hold true for video tapes, audio recordings and printed material. Supreme Court decisions have upheld the rights of the individual to copy material for his own use and for the nonprofit enjoyment of his friends.

A word of warning now. Just because you can make nonprofit copies, doesn't mean you have right to become the self-proclaimed east coast distributor for the software manufacturer of your choice. A civil suit brought by a software developer, manufacturer or distributor is not dependant on the profit motive, and civil cases need not be proven "beyond a reasonable doubt" but only "by a preponderance of evidence" which is much easier to prove. They must show that they received damages because of your actions. If you have been keeping your computer warm into the wee hours in order to make friends and influence people with free software, you may discover that this will be the easiest portion of their case. You may also discover that the software wasn't free after all. Paying direct damages is bad enough, but watch out for the punitive damages, the sometimes enormous fine which may also be required for you to pay to the plaintiff if you lose the suit. And don't forget, we haven't even discussed state criminal laws.

If you should take that one step too far, and violate the Federal law, who would investigate it (and possibly arrest you)? The Federal Bureau of Investigation has primary jurisdiction on Copyright violations. They share it with U.S. Customs and the Postal Service, depending on the method of operation of the violator. In major cities, the FBI has "White Collar Crime" squads which enforce the laws in question. That's right, back-up your software and you may be a "White Collar Criminal." That doesn't mean you're a

caucasian with a lapel fetish, but that Copyright violations are handled by the same squad that investigates embezzlement, contract fraud and political corruption. No wonder you don't see many piracy arrests, they've got their hands full now! I don't, however, recommend testing just how busy your local FBI field office may be. You may be surprised at their eagerness to give you all the attention you've richly deserved. By the way, the U.S. Courts system does have procedures for handling juveniles.

This article was written to calm the minds of those who make backups of software, or any other Copyrighted material, for their own use. It served, as a result, to define the separation between legal action and crime and possibly enlighten some who regularly copy software. Advising you about how a law reads is not a suggestion to break that law or to stretch it to its limits. Someone asked me to define the quantities or financial amounts which must normally be involved before the FBI would determine they would investigate or the U.S. Attorney's Office would seek prosecution. But that's where I must draw the line, since that information is only of real value to the true software pirate, and keep the answer our little secret. The information provided here can be found in any public library. The Federal Copyright laws can be found in Titles 17 and 18 of the United States Code. People who violate those Titles can be found in any Federal prison.

****

PROGRAMMING TIPS AND TRICKS

by Earl Hill

This month I am going to get into some graphics and a programming technique which we have not touched on before. This technique is called Page Flipping. It is a very useful technique and can give some striking visual effects. Also, it offers another means of animation. The sub-title of this month's column is "Page Flipping Color Graphics".

There have been several articles in magazines such as COMPUTE! on rapid page flipping of screens by display list modification - which is what page flipping really is. Some games use page flipping (for example see the article by David Plotkin in COMPUTE's Second Book of Atari Graphics, page 95) to instantly change the background independently of Player-Missile Graphics. Full screen animation is also possible.

Many of the page flipping programs were originally written as, for example, bulletin boards (BBS's) in GRAPHICS text modes 0, 1, 2. However, not entirely unexpectedly, serious problems can arise when you try to use the same programs in the regular GRAPHICS modes such as 3 or 5 or 7.

For the workable, foolproof solution to page flipping in the color GRAPHICS modes, lets briefly review page flipping in general. Page flipping relies on several important memory locations. The first one is locations 88 and 89 where the beginning of screen memory is normally contained. This two-byte register is used only for PRINTing and PLOTting to the screen. The other memory locations are the familiar address of the display list, locations 560 and 561. Both of these addresses are in lo- and hi-byte form. Incidentally, this column is 8-bit computer oriented, so all these programs and tips apply only to your Atari 800, XL's, XE's, etc., not to the ST's. These latter locations are concerned with screen display. Page flipping uses these two registers to draw exhibits at various places in memory, and then changes the display list register to instantly point to the various screens desired. We find the display list register by DLIST (or DL for short)=PEEK(560)+256*PEEK(561). Bytes five and six of the display list (DL+4,DL+5) contain the address of the screen memory. Locations 88 and 89 are used to point to the part of memory where your screens are stored. One last item before we move on to examining the program. To make it easy on ourselves and to simplify programming, we set the lo-byte of location 88 to zero. That way only the hi-byte or DL+5 needs adjusted to do our address changing (page flipping).

There are several approaches to programming for page flipping. As I said, many of these are written to work only in text modes such as GRAPHICS 0. Page flipping color graphics displays is not any more complicated if proper allowance is made for memory usage. GRAPHICS 5 is a good choice for color page flipping since it has four colors and uses only about 1K of memory per full screen. Now let's look at my example program. What we want to do is display four separate screens with a picture for each. Since for each GR.5 screen we require about 1K, we will need to step back from RAMTOP (location 106, remember the tutorial on P/M Graphics?) a total of 16 pages to give us room to draw our four pictures. In the program, this starts at line 805, where we also make our first graphics call.

Here is the entire program:

```
80 REM ** MULTIPLE PAGE FLIPPING IN COLOR
90 GOSUB 800
100 FOR I=1 TO 10
110 GOSUB 200
120 POKE DL5,SCR1:GOSUB 300
130 GOSUB 200
140 POKE DL5,SCR2:GOSUB 300
150 GOSUB 200
160 POKE DL5,SCR3:GOSUB 300
170 GOSUB 200
180 POKE DL5,SCR4:GOSUB 300
190 NEXT I
195 END
200 REM ** SOUND
210 SOUND 0,100,10,15:FOR X=1 TO 20:NEXT X
220 SOUND 0,0,0,0:RETURN
230 RETURN
300 REM ** DELAY
310 FOR DE=1 TO 500:NEXT DE:RETURN
400 REM ** ATARI SYMBOL SKETCH
405 COLOR 1
410 PLOT 37,4:DRAWTO 37,20:DRAWTO 25,38
420 PLOT 39,4:DRAWTO 39,38:PLOT 40,4:DRAWTO 40,38:PLOT 42,4:DRAWTO
42,20:DRAWTO 55,38
430 RETURN
500 REM ** PATTERN
505 COLOR 1:SETCOLOR 0,4,10
510 FOR RON=10 TO 40 STEP 2
520 COL=40:PLOT 0,10:DRAWTO COL,RON:PLOT 79,10:DRAWTO COL,RON
525 NEXT RON
530 RETURN
600 REM ** DRAW CIRCLE
610 COLOR 2:SETCOLOR 1,8,8:SZ=25
620 Y=SIN(T)*SZ*0.9:X=COS(T)*SZ
630 PLOT X+40,Y+24:FOR T=0 TO 179
640 Y=SIN(T)*SZ*0.9:X=COS(T)*SZ:Y=Y*0.95:DRAWTO X+40,Y+24:NEXT T
650 RETURN
700 REM ** DRAW STAR
710 COLOR 3:SETCOLOR 2,12,8
720 PLOT 26,40:DRAWTO 40,8:DRAWTO 54,40:DRAWTO 22,20:DRAWTO 58,20:DRAWTO
26,40
730 RETURN
800 REM ** SETUP STARTS HERE
805 RT=PEEK(106)-16:GRAPHICS 5+16
810 POKE 88,0:POKE 89,RT-4:? #6;CHR$(125)
815 POKE 106,RT
820 GRAPHICS 5+16
825 POKE 559,0
830 SCR1=RT:SCR2=RT+4:SCR3=RT+8:SCR4=RT+12
835 POKE 88,0:POKE 89,SCR1:GOSUB 400
840 POKE 88,0:POKE 89,SCR2:GOSUB 600
845 POKE 88,0:POKE 89,SCR3:GOSUB 500
850 POKE 88,0:POKE 89,SCR4:GOSUB 700
855 DL=PEEK(560)+PEEK(561)*256:DL4=DL+4:DL5=DL+5:POKE DL4,0:POKE DL5,RT+4
860 POKE 559,34
870 RETURN
```

The next thing we do is, in lines 810 and 815, POKE values into

MICRO LEAGUE SPORTS ASSOCIATION

JULY 10, 1987

DEAR MR. ADAMSON,

GREETINGS FROM THE COMMISSIONER'S OFFICE OF MLSA. THANK YOU FOR YOUR LETTER AND YOUR INTEREST IN MICRO LEAGUE WRESTLING.

WE ARE CONTINUING DEVELOPMENT AND TESTING OF OUR WRESTLING DISK. DUE TO A CHANGE IN SPECS AND OUR CONTINUED COMMITMENT TO QUALITY THE RELEASE DATE HAS BEEN DELAYED. OUR TARGET DATE IS EARLY SEPTEMBER OF THIS YEAR.

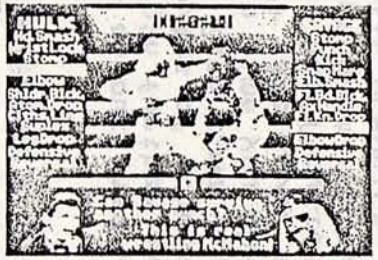
ENCLOSED, YOU WILL FIND SEVERAL FLYERS. I HOPE THAT THESE WILL BE OF HELP TO YOU.

THANKS AGAIN FOR YOUR INQUIRY. WOULD YOU PLEASE SEND US A COPY OF YOUR NEWSLETTER FOR FUTURE REFERENCE. THANK YOU FOR YOUR COOPERATION.

CORDIALLY

Tom Metallo

TOM METALLO
LEAGUE OFFICE STAFF



Be the foe and try to take Hulk's title away—the moves are up to you.

locations 88 and 89 corresponding to the reserved area of memory. The screen is also cleared at this time. Line 820 is a necessary second graphics call which is critical for correct positioning of the screens.

Since the PLOT and DRAWTO's in Basic take some time and we are going to do four screens, we turn off the video display in line 825. Then in line 830 we define our four screens as equates to RT. merely adding four for each (we stepped back 16 pages in line 805). In lines 835-850 we POKE memory location 88 with zero and location 89 to the four screen addresses where we draw our pictures. Have fun putting your own pictures here! However, as you know, the longer it takes to draw the picture the longer it will be before you have something to point to. Simple PLOT, DRAWTO pictures are best rather than involved graphics designs. Of course, don't go outside the screen limits of GRAPHICS 5 or you'll get an error message.

After drawing the screens with their respective colors, we locate the display list in line 855. The fifth and sixth bytes, as we said, contain the address of the display memory. Now we are ready to look at the screens in living color! First we turn the screen back on in line 860. Then, since the lo-byte has been set to zero, we only need to point to location 89 for our pictures. Pretty clever, huh? As each picture is flipped, we add a little sound and a small delay so that each picture can be seen. The FOR/NEXT loop cycles 500 times. Change this number to suit yourself - try it and see what happens.

A few final thoughts on the program. Of course, you can display any color or color combination you desire for each screen. Instead of a dark background, you could, for example, add a POKE 712,14 to line 820 for a white background (or any other color desired depending on the value POKEd). You can also change screens by various methods such as keyboard keys, joystick movement, or by LOCATEing screen coordinates (as in a maze game). Probably the most useful aspect of page flipping is changing backgrounds while using other means (such as P/M graphics) for moving objects or figures. The speed at which these changes are possible (you almost won't believe it!) can be seen by changing line 100 to: FOR I=100 TO 1 STEP -1 and line 310 to FOR DE=1 TO 5:NEXT DE:RETURN. Try this for yourself.

With program modification you can use other GRAPHICS modes. However, you are limited due to overall memory requirements. To see the effect of overlapping memory, try changing the graphics command in lines 805 and 820 to GR. 7+16. The result will be somewhat weird, but rather colorful. Of course, the Graphics mode can be changed to GR.4 (no memory problems); however, you will get fewer colors. Each major Graphics choice with differing screen coordinates will need to be taken into consideration in your programming. However, again, select a mode which is easy on memory and has a good color selection. GRAPHICS 5 as in this program is a good, all-purpose choice.

By following the same steps as outlined you can easily add more screens. Each new screen must begin further away in memory than the previous screen by the various bytes of screen memory required for each display, to prevent overlapping. Each time you RUN this program push RESET, not just BREAK, since each time you RUN it you are stepping back RAMTOP which could get your ATARI into never-never land. However, if you wish to RESET without pushing the RESET button, add line 193 A=USR(58484). Then just RUN the program. Within these constraints you can have some great fun, some fantastic graphics' displays and some super animation.

More On The ST-Transformer

EXCERPTS FROM COMPUTER SHOPPER AND MICHIGAN ATARI MAGAZINE

by John Nagy

The adventure of Darek Mihocka and his developing Atari 800-in-an-ST program continued this month with a surprise appearance and firery confrontation at the April Buffalo Atarifest. It culminated in what appears to be an agreement by Atari to allow distribution of the program.

The possible agreement followed months of controversy and pressure from Atari user groups begging for a way to link Atari 8-bit software with the new and much more capable 16-bit ST computers.

At the Buffalo Atarifest in late April, two user groups showed the "ST Transformer" in operation. The author sent both the Genesee Atari Group (GAG, Flint, Michigan) and the Westmoreland Atari Computer Organization (WACO, North Huntingdon, Pennsylvania) copies of the public domain emulator for demonstration only. Interest was brisk at both tables despite the uncomplete state of the program and the current slowness of execution (20%-40% "normal" speed depending, on the program). Atari tried to ignore the demos, but WACO members George Adamson and his son Jim kept at user group coordinator Sandy Austin and others for comment. They report that it turned into a h[illegible]g angry argument that i[illegible] from accusations of r[illegible] threats of "letting the l[illegible] decide."

WACO members quizzed Atari reps over the actual status of the Translator, which was sent to all registered user groups for free distribution. Sandy first said that it was not sent and remained Atari's property, then admitted that she didn't know for sure. At issue was what restrictions (if any) can now be put on the uses of the disks after years of free distribution. WACO reps offered to return Atari's Translater to Sandy while loudly decrying Atari user-group support. Sandy refused to take it back.

. Atari had not taken any action, or even any position regarding any of the "ROM Clones" until Darek's Transformer came on the scene. It appears that Atari would first have to legally assert ownership and control of all the "translators" in order to get any legal claim against Darek for using one or more of them in his 800 emulator. It would be a large and ugly task, and would surely darken the already black eye that Atari has garnered with their 8-bit owners and user groups.

The dark prospects of long and possibly unsuccessful legal action, together with petition drives, newsletter editorials, and comments in support of the ST Transformer—now [illegible] have gotten through to

[illegible]y has Atari been so hard o[illegible] this project from the beginning? No clear reason has been given, or at least not the same one twice. Although Atari may have had the right to prevent the use of their ROMs, that didn't make their prevention right. To his credit, Darek Mihocka has consistently tried to follow the law and Atari's wishes, and ceased distribution of his prototype emulator last fall when he first talked to Atari. Only after national attention to the situation has any progress been made.

Throughout the months of discussion on the subject, Neil Harris and company at Atari kept asking, "Why would anyone want to use 8-bit software on an ST?"... Perhaps a much better question is "WHY NOT?" WACO and other user groups want an emulator to provide some kind of link, however flawed, between the two products of Atari Corp. They point out that early press releases from Atari about what was then called "the 1985 project" all pushed great new 16-bit power but full compatibility with existing software. The Apple people accomplished this compatibility for Apple II users with the latecomer Apple IIgs. Obviously, some people at some places at some-times felt it was important.

FROM COMPUTAH

It's happened before. A group of disgruntled people have changed the whims of big business on many occasions.

Witness the rebirth of television shows such as Hill Street Blues, and Cagney and Lacey, all ressurected by viewers banding together for a common goal.

Or how about the recent actions of Springboard Software. They have recently released Certificate Maker for the ST, and versions of The Newsroom are in the works for both the 8-bit and the ST, all as a direct result of write-in campaigns from thousands of Atari owners.

The greatest triumph of all may well be a modern-day David and Goliath story: the case of the People vs. Atari Corporation, otherwise known as 'Transformer, Transformer, Who's got the Transformer.'

In an amazing grassroots movement, users, and user-groups across the country were banded together by a few members of the Westmoreland Atari Computer Organization (WACO), in Pennsylvania. Through the dedicated efforts of Larry "Blaze" Sabatino and George Adamson, the WACO folks organized petitions, a letter writing campaing, and made a great deal of noise over Atari's attempts to block the production of an 8-bit emulator for the ST. As a result, the little-emulator-that-wasn't is now the little-emulator-that-will.

So what does this all prove? We are all painfully aware of Atari's lack of concern for the future of the 8-bit computers. Likewise, Atari's earliest sentiments toward the ST Transformer were 'why would anyone want one?'; another don't-care attitude.

In a resounding show of solidarity, the user group community has given notice; 'we care about our computers, and if we want something bad enough, we can put the pressure on'.

Hard enough, evidently, for even Atari to feel the heat.

WHAT OTHER NEWSLETTERS SAID ABOUT WACO

EIGHT-BIT EMULATOR FOR ST: The program that almost wasn't

By Bob Buman FROM SAGE SCROLL

April and May have been HOT months concerning the topic of an 8-bit emulator; software which enables you to run Atari 800 (8-bit) software on the Atari ST-series computers. Due to legalities the program almost got buried, but thanks to strong efforts by WACO and other user-groups, the program is alive and well. Here's a list of the main players in this "show-down":

DAREK Mihocka: Programmer from Ontario, Canada; writer of the 8-bit emulator.

ATARI CORP.: Owner of the sacred 800 operating System, much needed for the emulator to work.

NEIL HARRIS: Public relations man at Atari Corp. The "man-in-the-middle" with the tough job of keeping Atari's image spotless through the battle.

JIM AND GEORGE ADAMSON & WACO: protectors of the emulator in U.S. territory; keepers of the faith (that great old Atari free spirit), and all-around good guys!

SUPPORTING ROLES: Michigan Atari user-groups

BATTLE GROUNDS: Buffalo Atari Faire and GENIE's Atari Roundtable.

OBJECTIVE: Secure the very existance of the 8-bit emulator for public use.

Why was Atari refusing to give Darek the rights to use the 800 operating system code? Wasn't that code released into the public domain when Atari released the XL-translator disk to user-groups for distribution? Was Atari purposely suppressing the emulator for some unknown reason?

Answers to these and other (very direct) questions were laced with heat and frustration by Mr. Harris. A registered letter was sent by WACO directly to Sam Tramiel concerning the whole matter. WACO distributed letters to a slew of user groups requesting each group to send petitions to Atari showing their support (that the emulator should be allowed to be produced and distributed).

This letter and the subject itself was discussed at our last meeting, but unfortunately the petition was misplaced.

Regardless, our group expressed support and many were anxious to sign the petition.

NEIL HARRIS QUOTED BY CIA ROM

NH: We've done the deal already with Darek. We wanted all along for Darek to release the source code to that because we felt that it's performance was poor and that the emulator was not really that worth while as it is now, but if the source code was out there, then there would be some enterprising folks who would be able to work some tricks and make it work better. He was reluctant at first, he has agreed to do that and we have agreed, in turn, to allow him to use our ROM code - our BASIC and Operating System together. So the next step is that he's looking for publisher, I think he's going to end up publishing it in ANALOG magazine and on their disk - probably in ST LOG I would imagine, and we have to actually sign a letter to make sure the agreement is formalized, but basically it's done. If you were a GEnie user, you could follow the progress of the negotiations on the whole thing because Darek went online and expressed dissatisfaction with the progress he was making with Atari and that was at the point that I went public and let people know where exactly we [Atari] stood on the matter. Due to user pressure, it all worked out.

A TOTALLY UNBIASED BENCHMARK COMPARISON, Part 1: (LDW did it!)

NOTE: These benchmarks were prepared by LDW and published in an (apparently uncopyrighted) ad on page 45 of the May, 1987 ST Applications magazine. We added DBASIC, of course.

BENCHMARKS:

A) 1M empty FOR/NEXT loops
B) Integer calculations (see listing B)
C) Float benchmark (see listing C)
D) Calc. standard BYTE magazine benchmark (May 85)
E) Sieve benchmark size 7000; determine first 1651 primes
F) Screen output 1000 strings of 70 characters (50 x 20 lines)

Speed comparison (all times in seconds)

	INTERACTIVE							
	rank	DBASIC	rank	GFA	rank	Fast	rank	NEW ST
A	1	17.6	2	48.1	3	66.0	4	303.0
B	1	50.1	3	527.0	2	526.0	4	1100.0
C	2*	10.0	3	10.2	1	6.0	4	15.5
D	1*	3.56	2	6.0	3	7.2	4	16.3
E	1	0.87	2	14.0	3	16.0	4	38.27
F	1	9.66	2	23.3	4	256.0	3	226.63

	INTERACTIVE		COMPILED							
	rank	DBASIC	rank	GFA	rank	LDW	rank	Philon	rank	SoftWorks
A	4	17.6	3	17.1	1	6.7	2	12.2	5	379.0
B	2	50.1	4	168.2	1	4.3	3	111.0	5	2542.0
C	3*	10.0	2	8.7	1	3.5	4	30.2	5	150.0
D	3*	3.56	2	3.5	1	2.8	4	8.2	5	22.0
E	1	0.87	2	1.3	4	1.8	3	1.5	5	33.6
F	1	9.66	3	21.9	2	13.9	4	58.2	5	62.1

	INTERACTIVE		COMPILED C's			
	rank	DBASIC	rank	Megamax	rank	Mark Williams
A	3	17.6	1	5.7	2	6.7
B	3	50.1	1	6.1	2	6.3
C	1*	10.0	3	58.8	2	37.2
D	1*	3.56	3	11.9	2	10.4
E	3	0.87	1	0.46	2	0.48
F	1	9.66	3	63.0	2	42.7

* DBASIC uses a double precision floating point format while the other BASICS use a single precision floating point format, except for GFA, which uses 1 1/2 (6 byte) precision.

BLOATED BEYOND BELIEF?

"Which is faster, an eleven year old 8-bit microprocessor, or a brand new state of the art 16-bit one with a high speed clock?

"The answer, of course, is that the old 8-bitter is significantly faster, because the programmers of any new chips always insist on using software and firmware that is hopelessly bloated beyond belief. As a general rule, any time you give a programmer a three times faster machine, they will write code that runs nine times slower than it did before." Don Lancaster, COMPUTER SHOPPER, Aug '87 p.145.

Gee, you don't suppose that Don was talking about GEM, do you?

Hal W Hardenbergh

Hal W. Hardenbergh
Chief Marketing Idiot
DTACK Grounded Inc

A TOTALLY UNBIASED BENCHMARK COMPARISON, Part 2: (I did it!)

NOTE: The following comparison of interactive DBASIC to compiled C is completely unbiased. I know; I wrote it, just as I wrote much of the DBASIC code.

Only a cynic would suspect that I might be biased.

The secret to winning at benchmarking is to make sure that the things your program is good at are, um, well-represented. For instance, the LDW BASIC compiler is very good at integer arithmetic, so three of the six benchmarks LDW used in their ad feature integers (A, B, and E). I can make DBASIC beat any ST C compiler 99 out of 100 tries; I can also make a good ST compiler beat DBASIC 99 out of 100 tries. If you wanta win you gotta choose your benchmarks good. Here, I am going to use a different approach, and examine *categories of performance*:

1) PORTABILITY: No contest. C wins hands down because C was *designed* for portability. After you finish writing your ST checkbook balancing program in C, you can easily port it to the Sinclair ZX-81, the IBM PC, the Tandy Color Computer, and the Cray 1. DBASIC, on the other hand, was *specifically written to run as fast as possible on the 68000 microprocessor* and hence is deliberately, as a fundamental design decision, not portable.

2) INTEGER ARITHMETIC: No contest. C wins again because it is a compiled language, capable of global optimization. Also C does not check for integer arithmetic overflows, which saves time (DBASIC does check). As a result, a good compiled C is nearly twice as fast DBASIC on the BYTE Sieve benchmark, which is a good all-around test of integer performance.

3) FLOATING POINT ARITHMETIC: No contest. DBASIC wipes out both the good ST compiled Cs in floating point. DBASIC wins both the speed contest and the accuracy contest. That's partly because the C language *requires, as part of the language specification,* that single precision arithmetic *be performed in double precision.* It is also partly because DBASIC has a six-year-old, and hence very mature, floating point math package.

4) STRINGS: This is a tough one. Computer experts agree that BASIC, in general, has much better built-in string-handling facilities than C. DBASIC, which emulates the industry-standard Bellevue strings, wins on convenience. On elapsed time? We've never seen a reasonably standard string benchmark. C might win, but the programmer would have to make a heroic effort because C simply does not have good built-in string capability. We're going to award this one to DBASIC.

5) DISK THROUGHPUT: No contest. DBASIC wins. All the ST compiled Cs use TOS because C is about *compatibility.* DBASIC is about *speed on the 68000 and ST,* and so uses a DOS which is nearly twice as fast as TOS.

6) INTERACTIVITY: No contest. DBASIC obviously wins because it is interactive; the ST compiled Cs obviously lose because they are not interactive. What could be simpler?

7) MATCHING ASSEMBLERS: Tie. Both DBASIC and the ST Cs include 68000 assemblers. (Ours is faster because it is written in assembly, but it's almost undocumented right now.)

8) SCREEN OUTPUT: We're including this for the same reason LDW did: we beat everything in sight! (LDW did not know about DBASIC.) But Megamax must have an awful lot of lead ballast in its screen output code; it is over six times slower than DBASIC. The Unicorn is over four times slower. Well, I told you that if you want to win, you have to choose the right benchmarks!

9) THE BOTTOM LINE: Obviously, the language you choose depends on how much time you want to spend learning it and what you want to do with it. I can't tell you whether DBASIC or a compiled C is better for you because I don't know what your needs are. I will say that, as interactive BASICs go, *DBASIC goes damn fast!*

DBASIC WILL NEVER RUN UNDER GEM! FORGET IT!

Us folks here at DTACK Grounded are simply amazed at the number of times you folks tell us, "Gee! That speed is simply great! Now give us that speed while we run TOS and GEM and play with our mouse!"

DBASIC is amazingly fast because it is totally independent of TOS, GEM, and small rodents.

GET YOUR COPY OF DBASIC WITH OFFICIAL LABEL AT WACO MEETING.

THE PRESIDENT'S PAGE

ATARI Planetarium, a Review

Hi there. This month I am pleased to do a review of the Planetarium program. Developed for ATARI by Deltron LTD., this program is for anyone who has ever looked up at the night sky and said, "I wonder what star that is." With its wide range of features it's designed to be simple enough for a child to use. Most features are accessed by the console keys and joystick/cursor keys with on-screen prompts to guide you. There are a couple of quirks that caused me some problems at first. Apparently the disk will only boot on an unenhanced 1050 drive. It won't boot on an 810 or a 1050 with the US Doubler installed. It will run only on XL/XE computers. (The other problem was getting a printout. More later.)

Once loaded the program defaults to the sky over Washington, D.C., Jan. 1, 1985. Enter the MAP mode and select your location on the world map by moving the cursor over the map or by setting the correct latitude and longitude. Next enter the SET mode and select the date and time for your sky show. Now select SKY mode and let the show begin.

After computing the location of the Sun, Moon and planets, a view of the sky about 72 degrees wide is displayed. To the right (not at the bottom) of this display is a text window showing technical data: time and date, elevation, azimuth, right ascension and declination. (These terms are explained in the manual but you don't have to understand them to use the program.) Also in the text window, the options are displayed. The viewing angle may set as narrow as 9 degrees for a closer look at a particular area. The time rate can be stopped or set to approximate real time. Or it can be stepped up 2 to 64 times faster than real time, forward or backward to see in minutes events that take hours in real time.

Still in the SKY mode, you may point the field of view to look in any direction by moving the cursor. You may also look directly north, east, south, west or opposite your current direction. But wait, there's more! You can get the program to display a field of view with a particular object centered: the sun, moon, any of the 8 planets, any of 88 constellations, or the north or south celestial pole. And you can lock on (TRACK option) to an object such as the moon, start the clock, and watch the sky move in relation to it, just as the real moon moves against the background of stars.

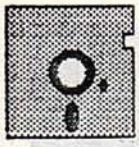
Some other options which can easily be toggled on or off in the display include:

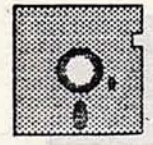
Imaginary lines in constellations
Abbreviated names of constellations
Symbols used to identify planets
Deep space objects (nebulae and galaxies).

Placing the cursor over a particular object and pressing the HELP key will bring up a description of the object from the data files on the back side of the disk. Brief descriptions of the 8 planets, 88 constellations, more than 1200 stars and more than 300 deep space objects are included. The sky can be viewed in any direction, even below the horizon and even with the sun in the sky. The sky is crosshatched to indicate below the horizon and shaded to differentiate between night (black), day (light blue) and various phases of twilight (deepening shades of blue). The twilight effect is also used during a solar eclipse.

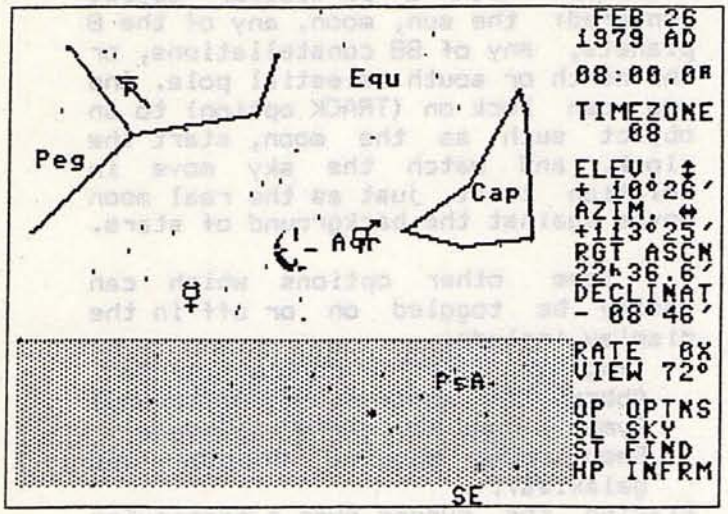
The CHART mode displays a view of the celestial sphere corresponding to the view in the SKY mode but with the horizon removed and north always toward the top of the display. Coordinate lines are displayed for right ascension and declination (think of longitude and latitude on earth's surface projected into the sky).

A hard copy can be printed if you have an Epson type or ATARI XMM801 printer. You may have to experiment to get a proper printout. On my SG10 I had to turn off the DIP switch for automatic line feed and select the XMM801 printer (it took me a while to figure this out.) Anything and everything displayed on the screen can





be printed, although printouts in SKY and CHART modes are the most practical (see illustration).



A TYPICAL VIEW IN SKY MODE. The bottom 1/3 of the display is below the horizon. The black crescent at the middle of the display represents the sun about 10 minutes before total eclipse. Also represented are Mars, Mercury, Pegasus, Capricorn and many stars. The arrow in the upper left points at the north celestial pole. Note on the right the time, date and other data.

The manual includes a Command Key Table or quick reference chart which concisely lists all commands. A brief but thorough discussion of the operating instructions is followed by several chapters which explain various aspects of astronomy. Also included are some technical notes on the program itself, several tables and a glossary. I found the manual to be as complete and easy to understand as any I have seen.

To test the program, I looked at a 1979 almanac to see what astronomical events occurred that year. I found that a total eclipse of the sun occurred on Feb. 26, 1979. It was visible over the eastern Pacific Ocean, Portland, Oregon, parts of Canada and Greenland. I set the Planetarium for the correct date and the longitude and latitude of Portland, Oregon. Starting the time at about 7:00 am, I watched as the eclipse commenced at about 7:20 am. The bright blue display began to darken until it turned dark grey with totality occurring about 8:10 am. Then the sky began to brighten as the eclipse faded, ending at about 9:00 am. (As mentioned in the manual, the positions of the sun and moon are updated every five minutes during an eclipse so a total eclipse of short duration may be hard to pinpoint. This particular eclipse lasted less than three minutes.)

It was refreshing to work with a program as complex as this one that does what it's supposed to do without running into bugs and without having to wade through uncomprehensible instructions. For ease of use, accuracy of documentation, practicality, you-name-it, this program rates tops. A must for any ATARI owner who has any interest in the night sky.

****

WACO Board of Directors

President: Chuck Cartia 678-2629
Past Pres.: Don Stoughton
Secretary: Debbie Davis 864-7216
Treasurer: Kurt Houghtaling 836-3653
Members at Large: Ed Reynolds
Bill Covert
Tim Enright
Newsletter Editor: George Adamson
Assistant Editor: Bud Burnside
8-bit Librarian: Don Shroyer 537-0452
8-bit Assistant: Keith Krause
ST Librarian: Blaze Sabitino 863-9436
ST Assistants: Rick Davis
George Adamson
Vice-president- Jim Adamson
Mailing Address:
230 Clairmont St.
North Huntington, PA 15642

Dues per year: $12/individual
$18/family & business

Meetings: 2nd Tues. each month at UP Church of the Covenant, Irwin, PA

Ad Rates: $20/page, $12/half page, $7/quarter page

WACO is an independent, non-profit Atari user group founded in 1983.

WESTMORELAND ATARI COMPUTER ORGANIZATION

SEPTEMBER 1987, LIBRARY DISK OF THE MONTH

SIDE ONE

BUDDAH.BIN---Graphics Buddah for you to chant to.

PISTONS.BIN--A neat graphics demo with animation. Looks foreign.

PWEEHER.MN---A sound and graphics Pee Wee Herman of your very own.

SIDE TWO

CONGA2--------Digitized sound and title Screen of ancient African Jungle Rythms. Interesting.

BACKNTME.DIG-Digitized Music.

LUCAS2.OBJ---Lucasfilms Demo. Graphics 8.

WOOFER.OBJ---No papers to clean up and no mouth to feed. The perfect watchdog.

YOUR ARTICLE WOULD HAVE LOOKED GREAT PRINTED HERE

W.A.C.O.-- HOME FOR EMULATORS

WACO's involvement in promoting 8-bit emulators for the ST gained attention in newsletters across the country (See reprints inside) and even in the COMPUTER SHOPPER.

While the world waited for Darek Mihocka's ST TRANSFORMER to be published in ST LOG, a software engineer from Mechanicsburg, PA, decided to not hide his light under a bushel basket any longer and announced plans for a 6502 cartridge emulator (using the Magic Sac concept-- You buy the board and plug in your own ROMS).

Steve Jones of Jonesware has promised to demonstrate his invention at the Tuesday, Sept. 8 meeting of WACO in the UP Church of the Covenant, Irwin, at 7:30 P.M. Other user groups have been invited. Since WACO was the first user group in the USA to see demos of the completed TRANSFORMER, it seems appropriate to have the opportunity to be among the first to see a hardware emulator in action.

Steve's cartridge emulates a 288K Axlon ramboard and will have its own cartridge port to piggyback 8-bit game cartridges. He hopes to have it available for the Christmas season at a cost of $50. Also, he plans special cables to connect 1050 drives to the ST. Eventually, the emulator should be capable of adapting MIDI and mouse operations.

Steve's friend Mike McCall is also expected to demo his Franklin (Apple compatible) cartridge which will allow Apple drives to interface with the ST, opening up thousands of educational programs (Schools, are you paying attention?) Some WACO members have wished for a Commodore 64 emulator to run all the C-64 software which was never converted to Atari 8-bit format. With the appropriate ROMS and a Commodore drive, Steve's future cartridge should make their wishes come true. Two other C-64 emulators are rumored to exist. One was seen demoed at the Atari booth at the CES.

****

Darek's TRANSFORMER will be published in the Sept. and Oct. issues of ST LOG.

****

NOTE- DTACK GROUNDED has given WACO its DBASIC language for the ST to distribute to all ST users. The company is attempting a novel marketing approach. If you like it, then you buy the thick manual. DTACK's own comparison of BASICS is inside. WACO will soon publish its own survey of popular ST BASICS.

WACO

WESTMORELAND ATARI COMPUTER ORGANIZATION

WESTMORELAND
ATARI COMPUTER
ORGANIZATION
230 Clairmont Street
North Huntingdon, PA
15642

S.L.C.C. JOURNAL

P.O. BOX 1506
SAN LEANDRO, CA
94577-0374